

PROGFUN-RAT1 Lenguaje funcional básico

Pregunta 1

Definiciones 01

En el lenguaje Haskell, ¿a qué se le llama una definición?

- ☐ A una asociación de un nombre (identificador) con un valor de un tipo particular.
- ☐ A la asociación de un nombre (identificador) con su tipo.
- ☐ A la declaración de una función.
- ☐ A la firma de una función.

Pregunta 2

Expresiones 01

¿Qué se entiende por expresión en un lenguaje funcional?

- ☐ Combinaciones de operaciones, funciones y valores que evalúan a un resultado.
- ☐ Cualquier combinación de símbolos que se puede escribir en el lenguaje.
- ☐ Cualquier valor que se le puede asignar un nombre.
- ☐ Definiciones de funciones cuyos tipos son coherentes.

Pregunta 3

Expresiones 02

Dada la siguiente función de Haskell: `square x = x * x` ¿cómo se podría evaluar la siguiente expresión?

- ☐ Todas las otras opciones son formas válidas de evaluar la expresión dada.
- ☐ `square (3+4) = square 7 = 7 * 7 = 49`
- ☐ `square (3+4) = (3+4) * (3+4) = 7 * (3+4) = 7 * 7 = 49`
- ☐ `square (3+4) = (3+4) * (3+4) = (3+4) * 7 = 7 * 7 = 49`

Pregunta 4

Expresiones 03

¿Cuál de las expresiones dadas es equivalente a la siguiente expresión de Haskell? `sqrt 2 * 2 - 1`.

- ☐ `(sqrt (2 * 2)) - 1`
- ☐ `sqrt (2 * (2 - 1))`
- ☐ `((sqrt 2) * 2) - 1`
- ☐ `sqrt ((2 * 2) - 1)`

Pregunta 5

Expresiones 04

Considere el siguiente expresión de Haskell: $2 * 3 + 4 * 5$ Indique cual de las siguientes es equivalente.

- ☐ $(2 * 3) + (4 * 5)$
- ☐ $2 * (3 + 4) * 5$
- ☐ $2 * (3 + (4 * 5))$
- ☐ $((2 * 3) + 4) * 5$

Pregunta 6

Expresiones 05

Considere el siguiente expresión de Haskell: $2 * 3 + 4 * 5$ Indique cuál es el la reducción correcta.

- ☐ $2 * 3 + 4 * 5 = 6 + 4 * 5 = 10 * 5 = 50$
- ☐ $2 * 3 + 4 * 5 = 2 * 7 * 5 = 14 * 5 = 70$
- ☐ $2 * (3 + (4 * 5)) = 2 * 3 + 4 * 5 = 2 * 3 + 20 = 2 * 23 = 46$
- ☐ $2 * 3 + 4 * 5 = 6 + 4 * 5 = 6 + 20 = 26$

Pregunta 7

Expresiones 06

¿Cuál de las siguientes condiciones es cierta?

- ☐ `"abcd" < "abcabc"`
- ☐ `"a" == 'a'`
- ☐ `"" == "`
- ☐ `"abc" < "abcd"`

Pregunta 8

Expresiones 07

¿Cuál de las siguientes condiciones es cierta?

- ☐ `'False' == "False"`
- ☐ `"True" == True`
- ☐ `False < True`
- ☐ `1 == True`

Pregunta 9

Expresiones 08

Indique cual de las siguientes expresiones de Haskell da 4.

- ☐ $((-4) \text{ `mod` } 3) + (8 \text{ `div` } 2)$
- ☐ $4 \text{ `mod` } (5 \text{ `div` } 2)$
- ☐ $((-4) \text{ `mod` } 3) + (4 \text{ `div` } 2)$
- ☐ $(4 \text{ `mod` } 3) + (5 \text{ `div` } 2)$

Pregunta 10

Expresiones 09

Considere que `||` y `&&` son los operadores lógicos de disyunción y conjunción convencionales de Haskell, y que `not` es la negación. Entonces, ¿cuál de las siguientes expresiones evalúa al valor booleano verdadero?

- ☐ `False && not True`
- ☐ `not False || True`
- ☐ `not True || False`
- ☐ `not False && not True`

Pregunta 11

Funciones 01

Por aplicación de una función se entiende:

- ☐ Chequear si los tipos de los argumentos son los que espera la función.
- ☐ Darle valores de entrada para obtener una salida.
- ☐ Obtener la cantidad de argumentos que tiene.
- ☐ Dar la definición de una función.

Pregunta 12

Tipos de dato 01

¿Qué entendemos por tipo de dato?

- ☐ Una colección de valores que se consideran juntos porque sobre ellos se pueden aplicar las mismas operaciones.
- ☐ Una marca que se les pone a las operaciones y funciones para poder verificar su coherencia.
- ☐ Una componente de las firmas que se utiliza para resolver la sobrecarga de funciones.
- ☐ Un conjunto de operaciones y funciones que tienen algún aspecto en común.

Pregunta 13

Tipos de dato 02

¿Cómo se escribe en Haskell el valor booleano falso?

- ☐ `False`
- ☐ `no`
- ☐ `0`
- ☐ `false`

Pregunta 14

Tipos de dato 03

¿Cómo se convierte un valor cualquiera a `String` en Haskell?

- ☐ En Haskell no existe el tipo de dato `String`.
- ☐ Con el método `toString()`.
- ☐ Con la función `show`.
- ☐ Concatenando el valor con el string vacío `""`.

Pregunta 15

Tipos de dato 04

Considere el siguiente código Haskell: `valor = 2 * 3 + 4 * 5` ¿Cuál es el tipo de valor?

☐ num

☐ number

☐ int

☐ Int